AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1-9. (canceled)

10. (currently amended) A method in a processor, in which data is processed in a pipelined manner, the data being included in a plurality of contexts, comprising a first context (3), each context passing a plurality of consecutive stages (2a-2f), in addition to which a plurality of operations is adapted to be executed on the contexts, each operation comprising a plurality of consecutive operation steps and the consecutive operation steps of one operation being executed on a context at least two different consecutive stages (2a-2f), the method comprising:

at a first stage (2a), executing an initial operation step (6a) of a first operation on the first context (3), and

at a second stage (2b) that consecutively follows the first stage (2a), subsequently commencing an execution on the first context of an initial operation step (7a) of a second operation before an execution on the first context (3) of a following operation step (6b) of the first operation is completed, ~~characterized in that~~wherein,

– at each clock cycle of the processor, the first context (3) is received at one of the stages from the preceding stage, the first context is unconditionally moved to a next stage and a subsequent context of a subsequent operation is received at the first stage (2a),

~~– the initial operation step (6a) of the first operation is executed on the first context (3) at a first stage (2a),~~

~~– the following operation step (6b) of the first operation is executed on the first context (3) at a second stage (2b), and~~

~~– the initial operation step (7a) of the second operation is executed on the first context at the second stage (2b)~~.

11. (previously presented) A method according to claim 10, comprising commencing at the first stage (2a) an execution of the initial operation step (6a) of the first operation on a second context before the execution on the first context (3) of the following operation step (6b) of the first operation is completed.

12. (previously presented) A method according to claim 10, comprising receiving at the second stage a result (R6a) of an execution of the initial operation step (6a) of the first operation.

13. (previously presented) A method according to claim 10, whereby at least one of the operation steps of the second operation comprises at least two alternative execution paths, and at least two of the alternative execution paths of the operation step are executed.

14. (previously presented) A method according to claim 13, further comprising:

- obtaining results (R7b1, R7b2) of at least two of the executions of the alternative execution paths, and

- determining, based on a result (R6) of an execution of an operation step of an operation initiated before the initiation of the second operation, which one of the results (R7b1, R7b2), of the executions of the alternative execution paths, an execution of an operation step of the second operation, following said operation step comprising at least two alternative execution paths, is to be based on.

15. (previously presented) A method according to claim 10, whereby the processor is arranged so that the following operation step (6b) of the first operation is presented to a programmer as being executed at the first stage (2a).

16. (previously presented) A method according to claim 10, wherein the first operation comprises a partial operation of executing (6c1) an instruction and a partial operation of writing (6c2) a result of the said instruction execution into a destination in a register, and the second operation comprises the partial operation of fetching (7a2 I, 7a22) an operand, the method comprising

(a) determining if a position in the register, from which the operand is to be fetched (7a2 1, 7a22) in the second operation, is identical with the destination of the partial operation, of the first operation, of writing (6c2) a result,

(b) if the result of the determination in step (a) is negative, fetching (7a21) the operand from the register, and

(c) if the result of the determination in step (a) is positive, fetching (7a22) the result of the said instruction execution.